

SEMANTIC WEB TECHNOLOGIES I

Lehrveranstaltung im WS09/10

M.Sc. Markus Krötzsch
Dr. Sebastian Rudolph

SEMANTIK VON RDF(S)

M.Sc. Markus Krötzsch

Einleitung und XML

Einführung in RDF

RDF Schema

Logik - Grundlagen

Semantik von RDF(S)

OWL - Syntax und Intuition

OWL - Semantik und Reasoning

OWL 2

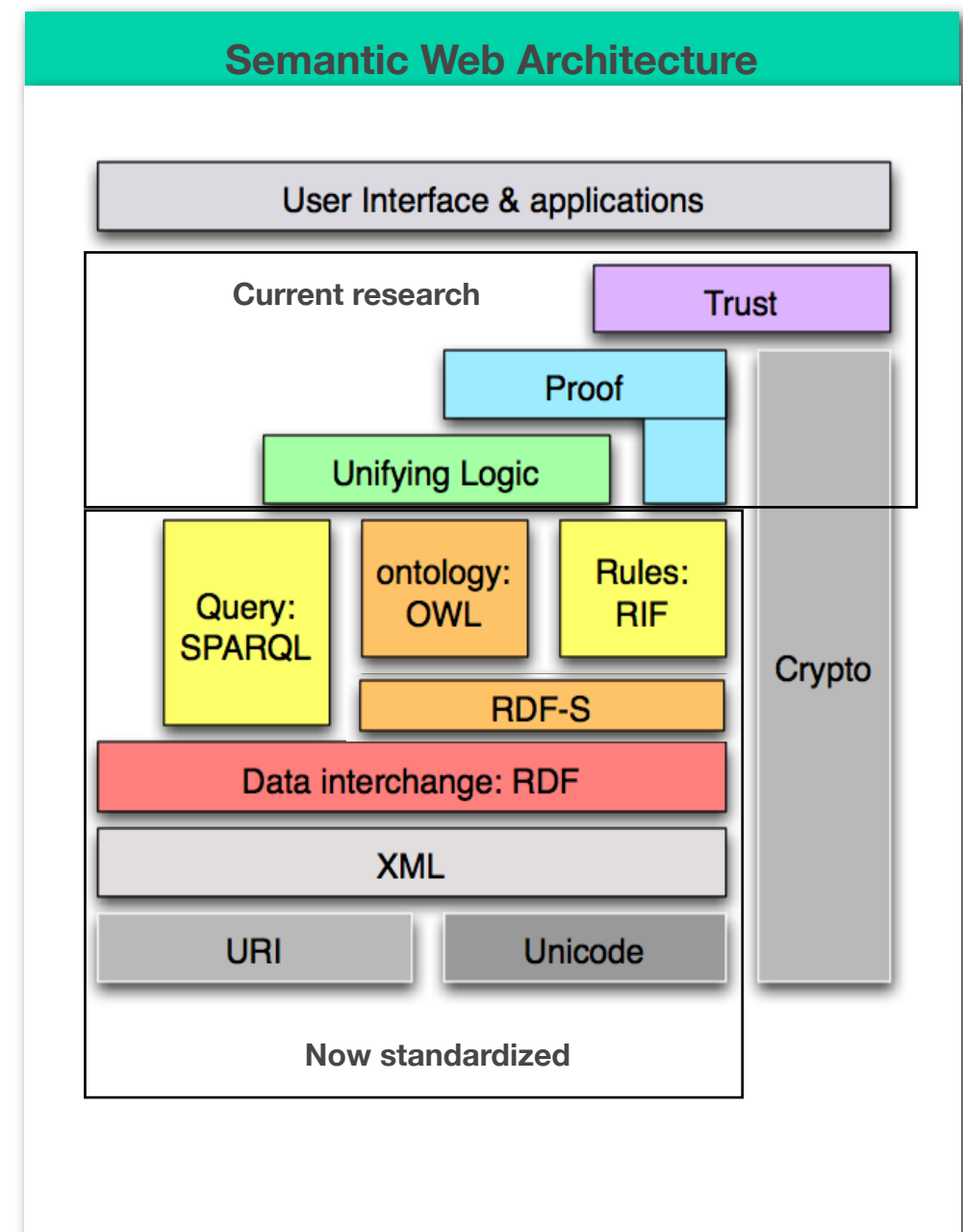
SPARQL - Syntax und Intuition

Konjunktive Anfragen / Einführung Regelsprachen

Regeln für OWL

Ontology Engineering

Semantic Web - Anwendungen



SEMANTIK VON RDF(S)

M.Sc. Markus Krötzsch

Einleitung und XML

Einführung in RDF

RDF Schema

Logik - Grundlagen

Semantik von RDF(S)

OWL - Syntax und Intuition

OWL - Semantik und Reasoning

OWL 2

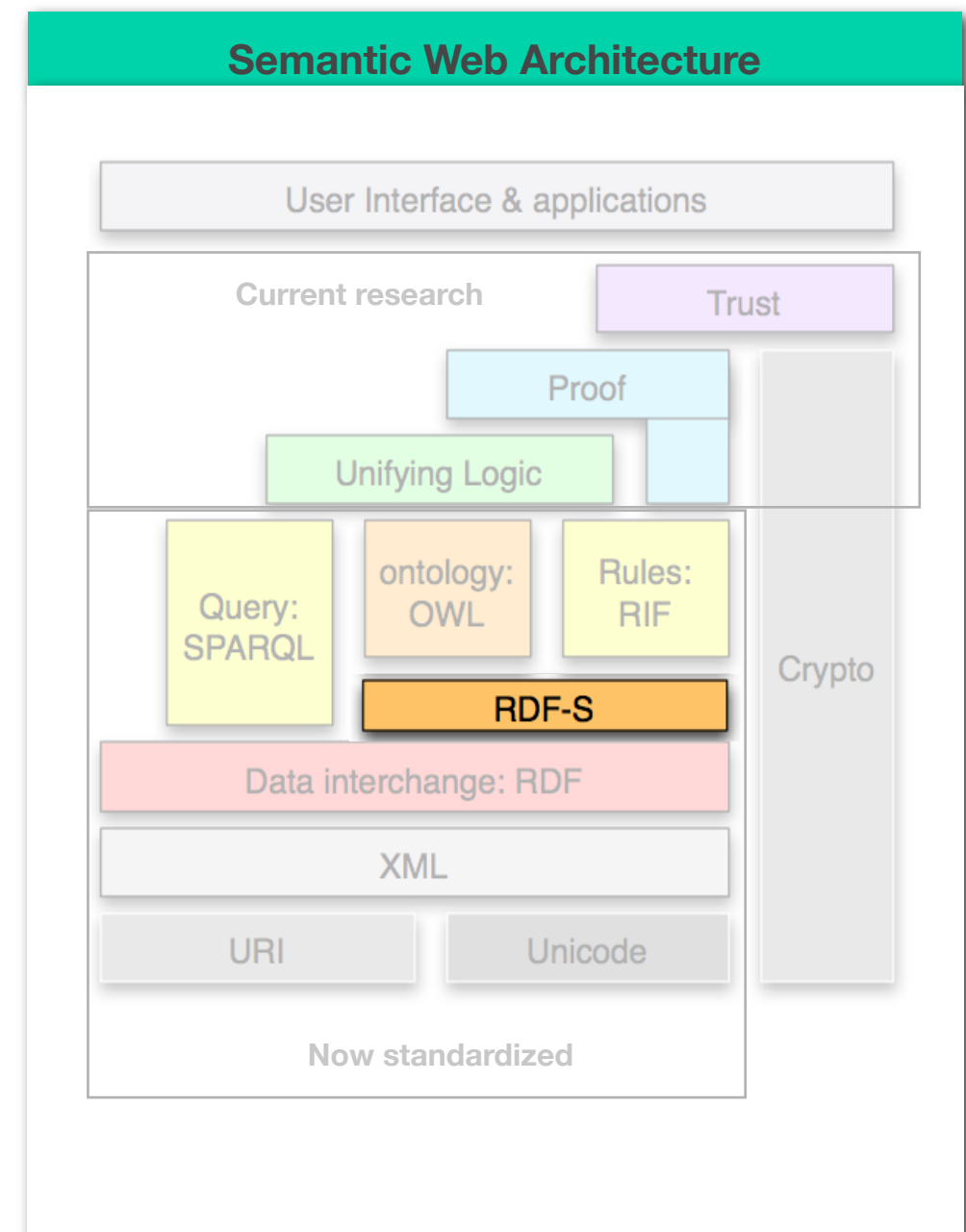
SPARQL - Syntax und Intuition

Konjunktive Anfragen / Einführung Regelsprachen

Regeln für OWL

Ontology Engineering

Semantic Web - Anwendungen



AGENDA

AIFB 

- Motivation
- Vorbetrachtungen
- einfache Folgerung
- RDF-Folgerung
- RDFS-Folgerung
- Unzulänglichkeiten von RDF(S)

AGENDA



- Motivation
- Vorbetrachtungen
- einfache Folgerung
- RDF-Folgerung
- RDFS-Folgerung
- Unzulänglichkeiten von RDF(S)

WARUM FORMALE SEMANTIK?

- nach Einführung von RDFS Kritik von Tool-Herstellern: verschiedene Tools - Inkompatibilitäten (trotz Spezifikation)
- z.B. bei triple stores:
 - gleiches RDF-Dokument
 - gleiche SPARQL-Anfrage
 - verschiedene Antworten
- daher: modelltheoretische Semantik für RDF(S)

AGENDA



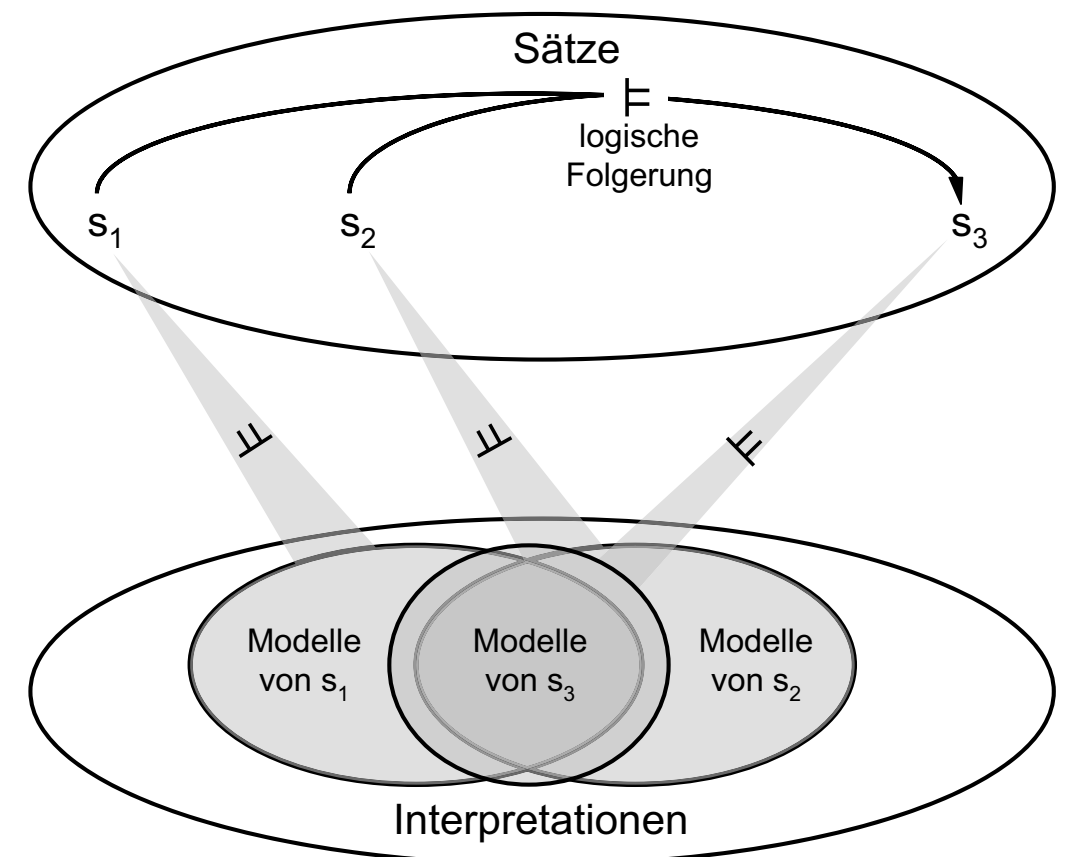
- Motivation
- Vorbetrachtungen
- einfache Folgerung
- RDF-Folgerung
- RDFS-Folgerung
- Unzulänglichkeiten von RDF(S)

WAS IST DIE SYNTAX?

- also: was sind die Sätze in RDF(S)?
 - Grundelemente (Vokabular V): URIs, bnodes und Literale (sind selbst keine Sätze)
 - jedes Tripel
$$(s,p,o) \in (\text{URI} \cup \text{bnode}) \times \text{URI} \times (\text{URI} \cup \text{bnode} \cup \text{Literal})$$
ist ein Satz
 - jede endliche Menge von Tripeln (genannt Graph) ist ein Satz

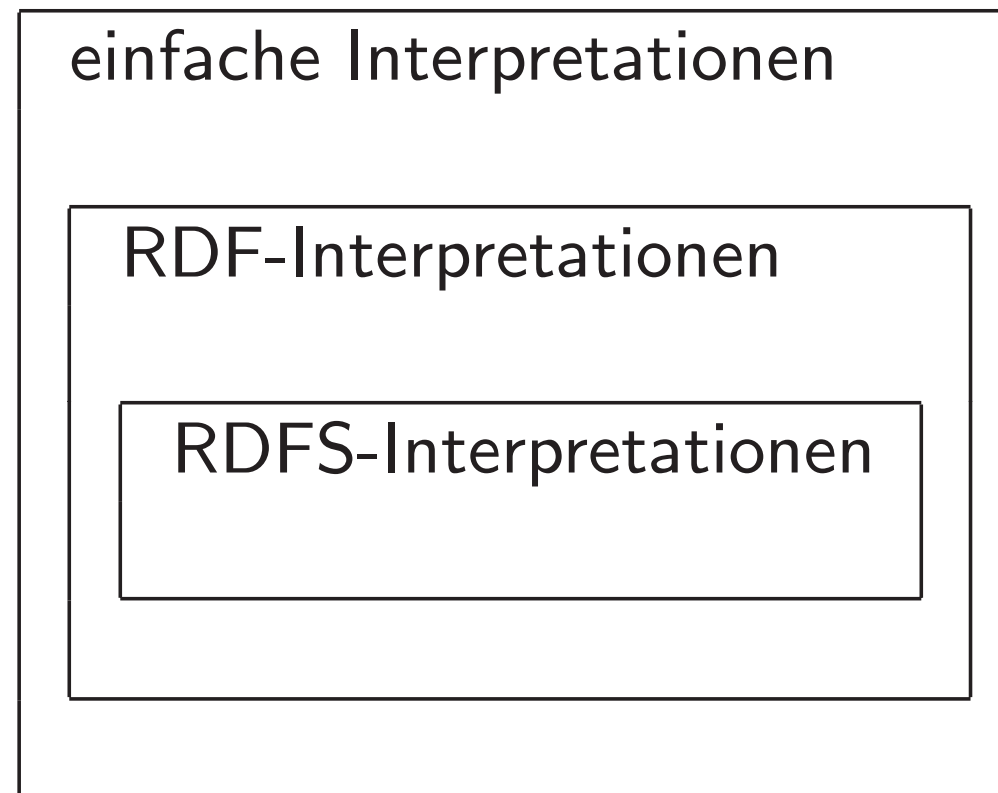
WAS IST DIE SEMANTIK?

- Konsequenzrelation, die sagt, wann ein RDF (S)-Graph G' aus einem RDF(S)-Graphen G folgt, d.h. $G \models G'$
- Modelltheoretische Semantik: wir definieren Menge von Interpretationen und legen fest, wann eine Interpretation Modell eines Graphen ist



WAS IST DIE SEMANTIK?

- Vorgehen schrittweise:



- je eingeschränkter die Interpretationen umso stärker die Folgerungsrelation

AGENDA



- Motivation
- Vorbetrachtungen
- einfache Folgerung
- RDF-Folgerung
- RDFS-Folgerung
- Unzulänglichkeiten von RDF(S)

SEMANTIK DER EINFACHEN FOLGERUNG

- einfache Interpretation:

Wir definieren also: Eine *einfache Interpretation* \mathcal{I} für ein Vokabular V besteht aus

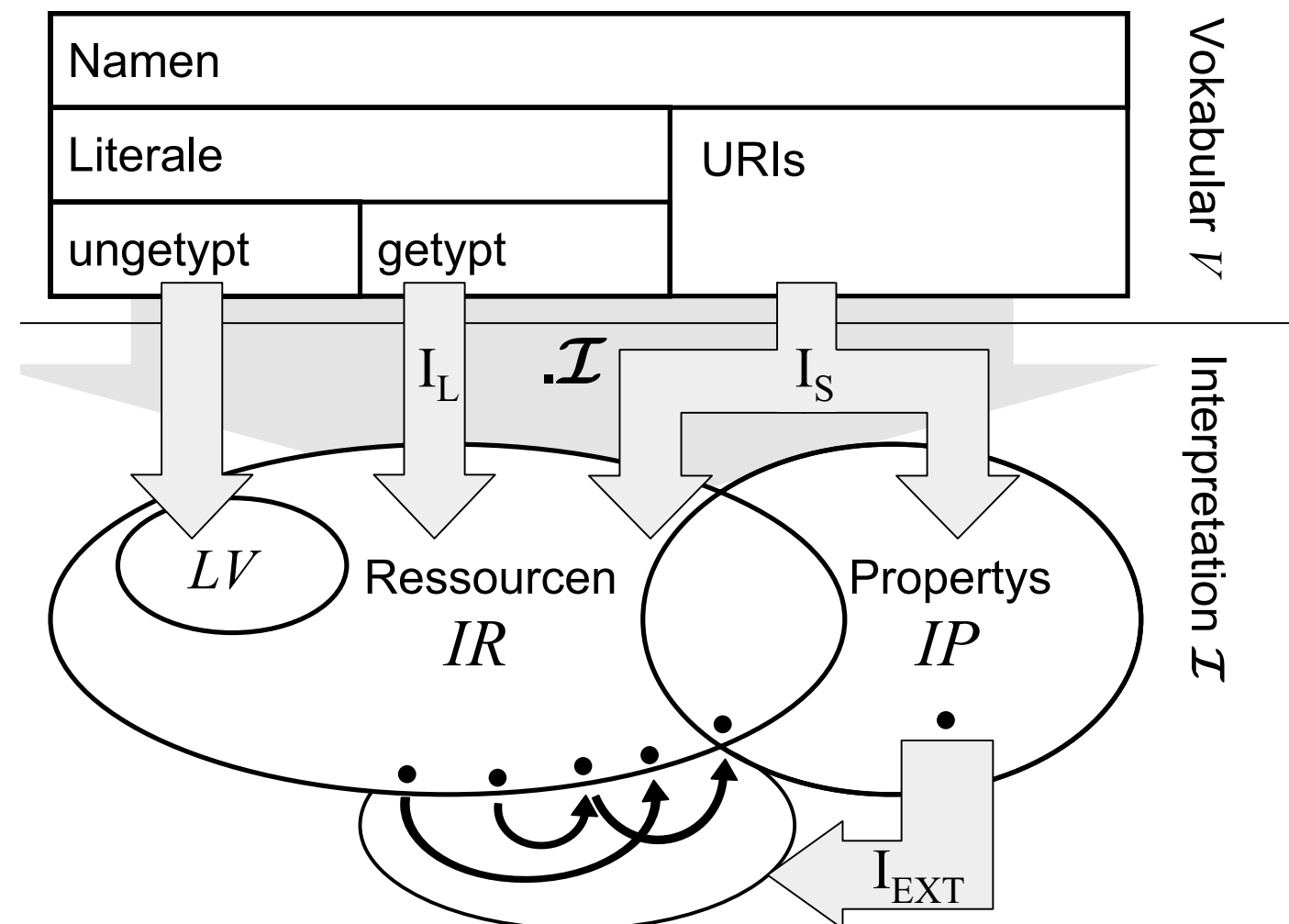
- IR , einer nichtleeren Menge von *Ressourcen*, auch genannt Domäne oder (Diskurs-)Universum von \mathcal{I} ,
- IP , der Menge der *Propertys* von \mathcal{I}
- I_{EXT} , einer Funktion, welche jeder Property eine Menge von Paaren aus IR zuordnet, also $I_{\text{EXT}} : IP \rightarrow 2^{IR \times IR}$, dabei nennt man $I_{\text{EXT}}(p)$ auch die *Extension* der Property p ,
- I_S , einer Funktion, welche URIs aus V in die Vereinigung der Mengen IR und IP abbildet, also $I_S : V \rightarrow IR \cup IP$,
- I_L , einer Funktion von den getypten Literalen aus V in die Menge IR der Ressourcen und
- LV einer speziellen Teilmenge von IR , genannt Menge der Literalwerte, die (mindestens) alle ungetypten Literale aus V enthält.

SEMANTIK DER EINFACHEN FOLGERUNG

AIFB

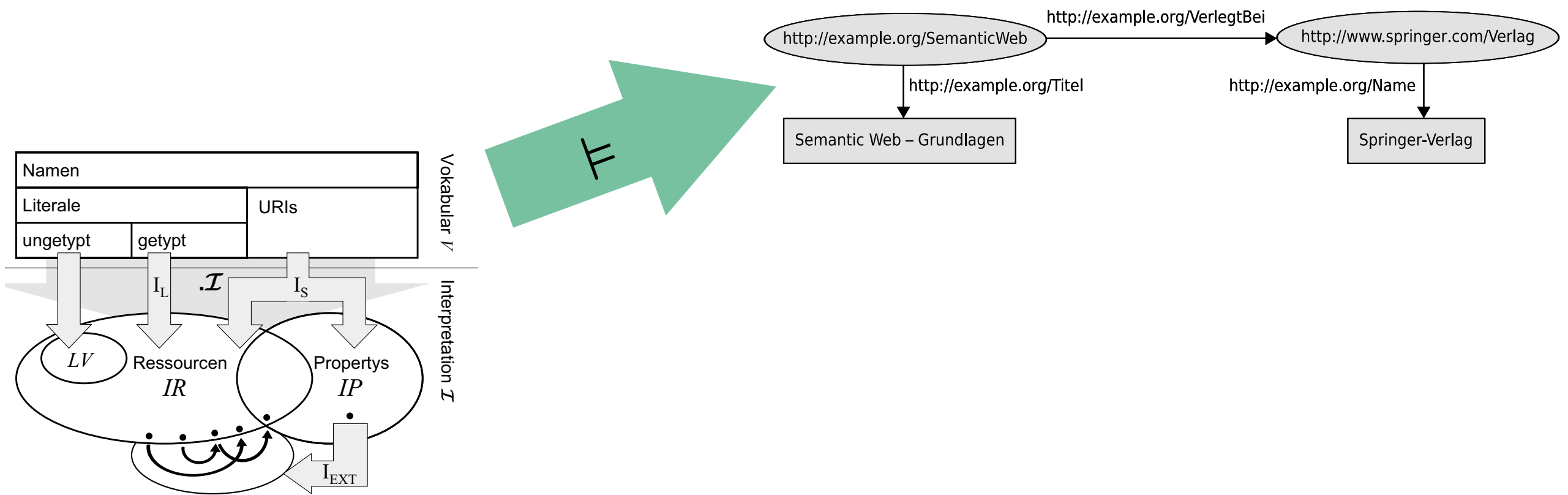
- jedes ungetypte Literal " a " wird auf a abgebildet: $("a")^{\mathcal{I}} = a$,
- jedes ungetypte Literal mit Sprachangabe " a "@ t wird auf das Paar $\langle a, t \rangle$ abgebildet: $("a"@t)^{\mathcal{I}} = \langle a, t \rangle$,
- jedes getypte Literal l wird auf $I_L(l)$ abgebildet: $l^{\mathcal{I}} = I_L(l)$ und
- jede URI u wird auf $I_S(u)$ abgebildet: $u^{\mathcal{I}} = I_S(u)$.

• Interpretation (schematisch):



SEMANTIK DER EINFACHEN FOLGERUNG

- Frage: Wann ist eine gegebene Interpretation Modell eines Graphen?

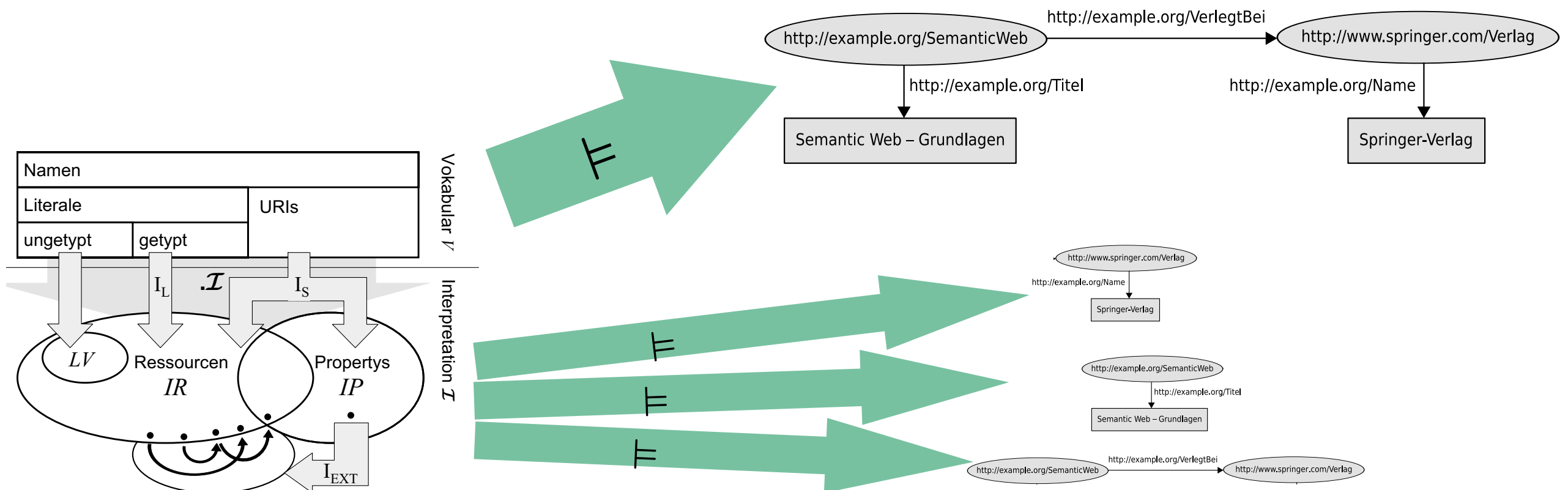


- ...wenn sie Modell jedes Tripels des Graphen ist!

SEMANTIK DER EINFACHEN FOLGERUNG



- Frage: Wann ist eine gegebene Interpretation Modell eines Graphen?

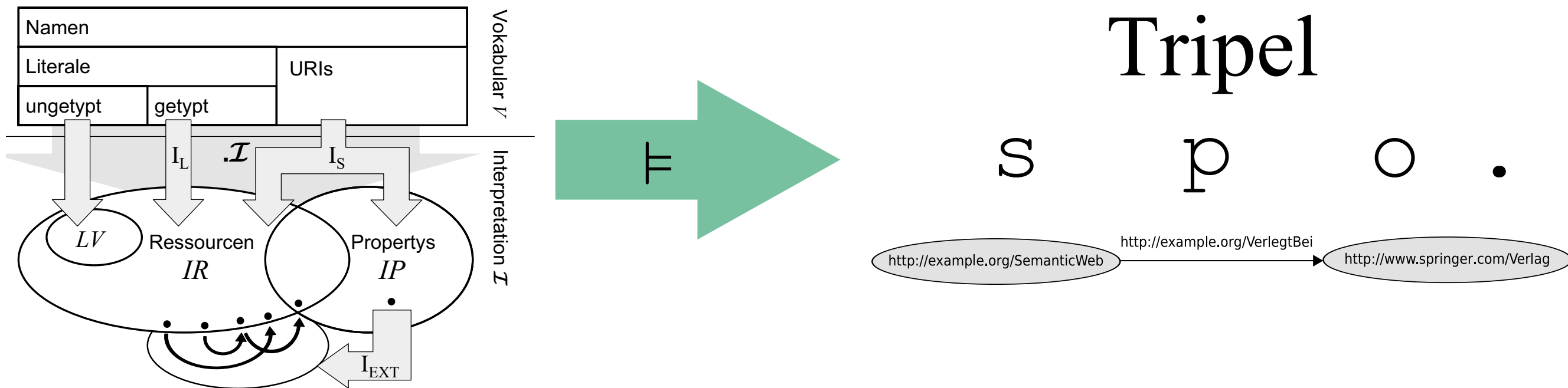


- ...wenn sie Modell jedes Tripels des Graphen ist!

SEMANTIK DER EINFACHEN FOLGERUNG



- Frage: Wann ist eine gegebene Interpretation Modell eines Tripels?

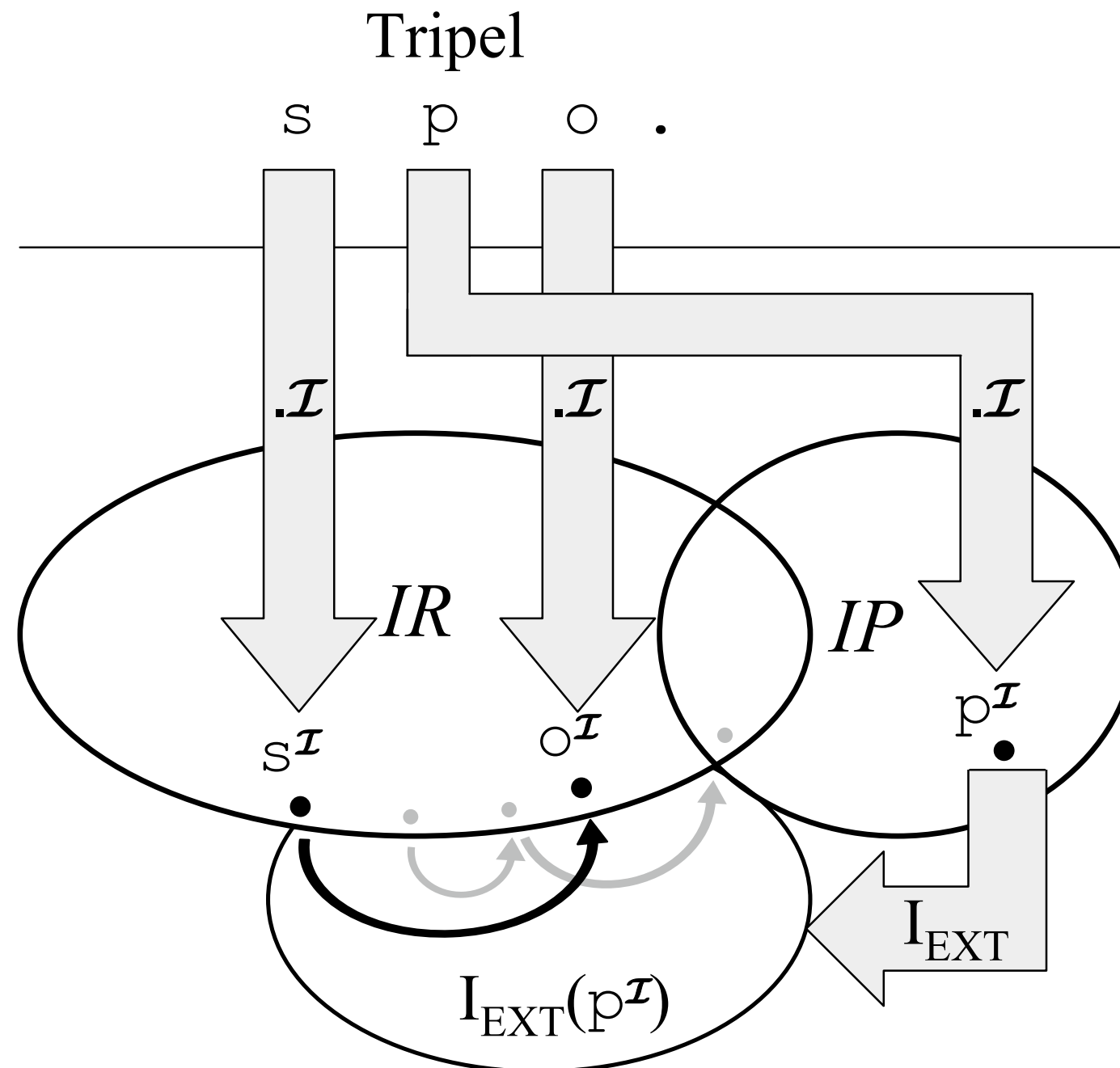


- ...wenn Subjekt, Prädikat und Objekt in V enthalten sind und außerdem:

$$\langle s^{\mathcal{I}}, o^{\mathcal{I}} \rangle \in I_{EXT}(p^{\mathcal{I}})$$

SEMANTIK DER EINFACHEN FOLGERUNG

- schematisch:



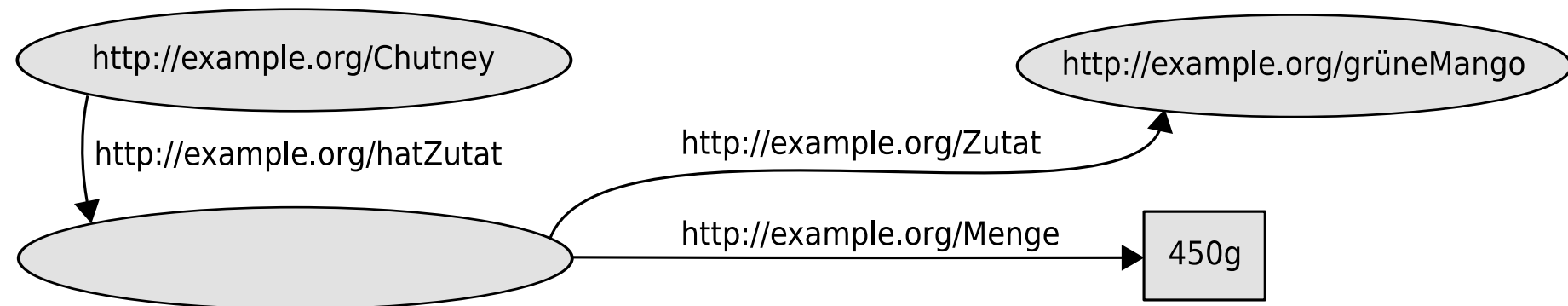
SEMANTIK DER EINFACHEN FOLGERUNG



- ...ups, wir haben die bnodes vergessen!
- wird nachgeholt: sei A Funktion, die alle bnodes auf Elemente von IR abbildet
- für eine Interpretation I , sei $I+A$ wie I , wobei zusätzlich für jeden bnode b gilt $b^{I+A} = A(b)$
- eine Interpretation I ist nun Modell eines RDF-Graphen G , wenn es ein A gibt, so dass alle Tripel bezüglich $I+A$ wahr werden

EINFACHE INTERPRETATION: BEISPIEL

- gegeben Graph G :



und Interpretation I :

$$IR = \{\chi, \nu, \tau, \nu, \epsilon, \iota, 450g\}$$

$$IP = \{\tau, \nu, \iota\}$$

$$LV = \{450g\}$$

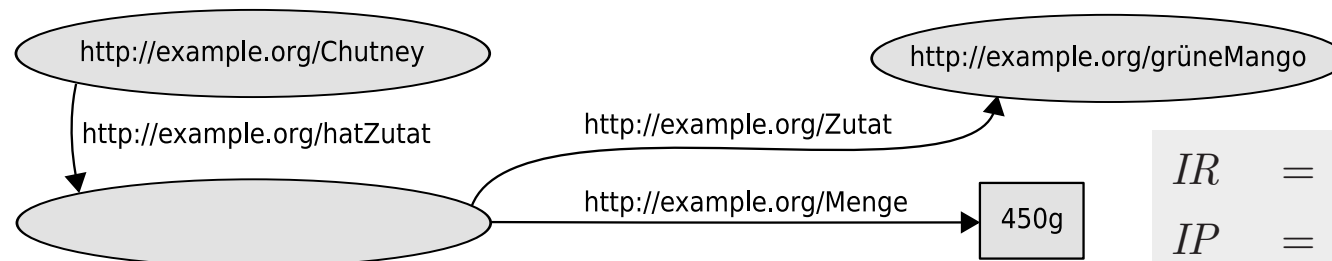
$$I_{EXT} = \begin{aligned} \tau &\mapsto \{\langle \chi, \epsilon \rangle\} \\ \nu &\mapsto \{\langle \epsilon, \nu \rangle\} \\ \iota &\mapsto \{\langle \epsilon, 450g \rangle\} \end{aligned}$$

$$I_S = \begin{aligned} ex:Chutney &\mapsto \chi \\ ex:grüneMango &\mapsto \nu \\ ex:hatZutat &\mapsto \tau \\ ex:Zutat &\mapsto \nu \\ ex:Menge &\mapsto \iota \end{aligned}$$

I_L ist die „leere Funktion“, da es keine getypten Literale gibt.

- ...ist I ein Modell von G ?

EINFACHE INTERPRETATION: BEISPIEL



$IR = \{\chi, v, \tau, \nu, \epsilon, \iota, 450g\}$
 $IP = \{\tau, \nu, \iota\}$
 $LV = \{450g\}$
 $I_{EXT} = \begin{aligned} \tau &\mapsto \{\langle \chi, \epsilon \rangle\} \\ \nu &\mapsto \{\langle \epsilon, v \rangle\} \\ \iota &\mapsto \{\langle \epsilon, 450g \rangle\} \end{aligned}$

$I_S = \begin{aligned} \text{ex:Chutney} &\mapsto \chi \\ \text{ex:grüneMango} &\mapsto v \\ \text{ex:hatZutat} &\mapsto \tau \\ \text{ex:Zutat} &\mapsto \nu \\ \text{ex:Menge} &\mapsto \iota \end{aligned}$
 I_L ist die „leere Funktion“, da es keine getypten Literale gibt.

- wählt man $A : _ : id1 \mapsto \epsilon$,
dann ergibt sich

$\langle \text{ex:Chutney}^{\mathcal{I}+A}, _ : id1^{\mathcal{I}+A} \rangle = \langle \chi, \epsilon \rangle \in I_{EXT}(\tau) = I_{EXT}(\text{ex:hatZutat}^{\mathcal{I}+A})$
 $\langle _ : id1^{\mathcal{I}+A}, \text{ex:grüneMango}^{\mathcal{I}+A} \rangle = \langle \epsilon, v \rangle \in I_{EXT}(\nu) = I_{EXT}(\text{ex:Zutat}^{\mathcal{I}+A})$
 $\langle _ : id1^{\mathcal{I}+A}, "450g"^{\mathcal{I}+A} \rangle = \langle \epsilon, 450g \rangle \in I_{EXT}(\iota) = I_{EXT}(\text{ex:Menge}^{\mathcal{I}+A})$

- also ist \mathcal{I} Modell von G

EINFACHE FOLGERUNG

- Definition der einfachen Interpretation legt (modelltheoretisch) einfache Folgerung für RDF-Graphen fest
- Frage: wie lässt sich diese (abstrakt definierte) Semantik im Sinne des automatischen Schlussfolgerns umsetzen
- Antwort: Ableitungsregeln

EINFACHE FOLGERUNG

- Ableitungsregeln für einfache Folgerung:

$$\frac{u \quad a \quad x \quad .}{u \quad a \quad _ : n \quad .} \text{ se1}$$

$$\frac{u \quad a \quad x \quad .}{_ : n \quad a \quad x \quad .} \text{ se2}$$

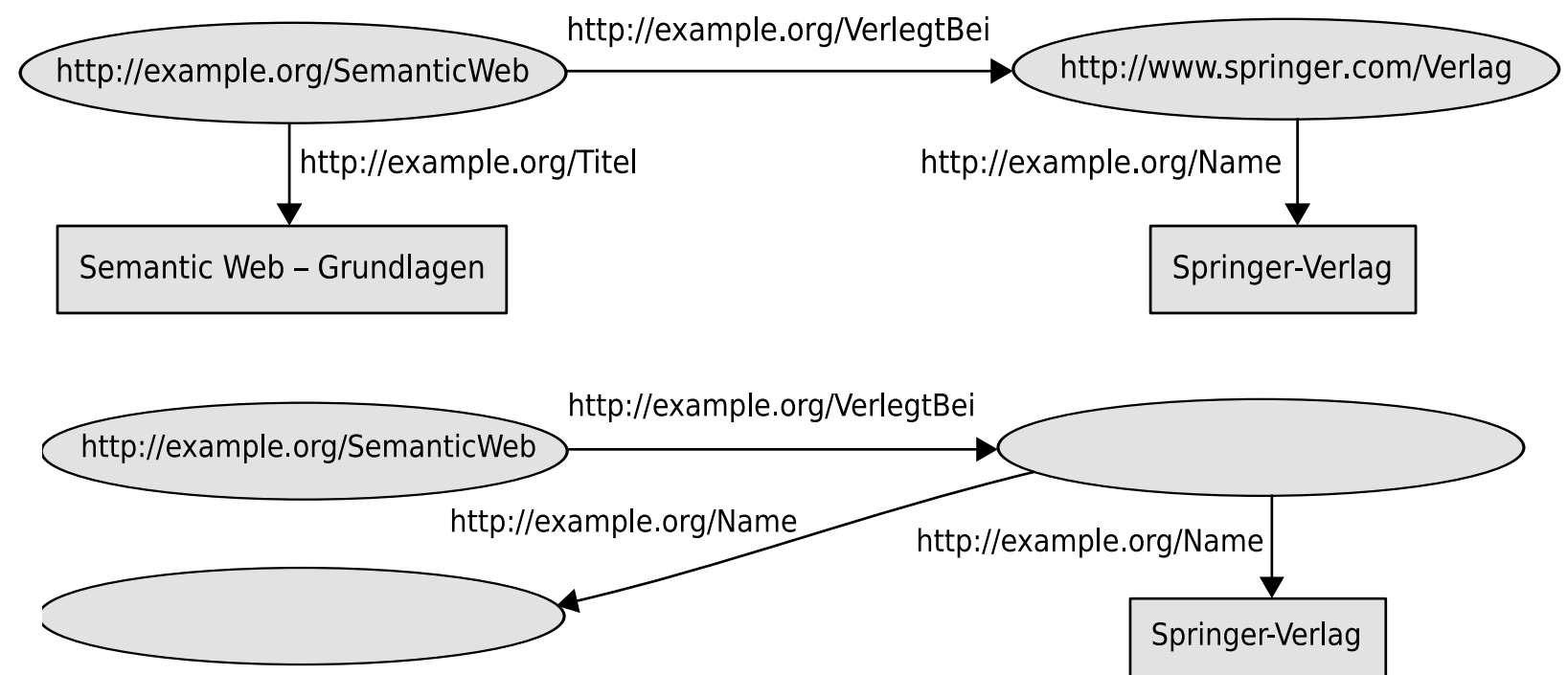
- Bedingung für Anwendung: leerer Knoten nicht bereits anderer URI / anderem Literal zugeordnet

EINFACHE FOLGERUNG

- Satz: Ein Graph G_2 folgt einfach aus einem Graphen G_1 , wenn G_1 mithilfe der Regeln se1 und se2 zu einem Graphen G_1' ergänzt werden kann, so dass G_2 in G_1' enthalten ist.

- Bsp.: aus

folgt ein-
fach



AGENDA



- Motivation
- Vorbetrachtungen
- einfache Folgerung
- **RDF-Folgerung**
- RDFS-Folgerung
- Unzulänglichkeiten von RDF(S)

RDF-INTERPRETATIONEN

einfache Interpretationen

RDF-Interpretationen

RDFS-Interpretationen

- ...RDF-Interpretationen sind spezielle einfache Interpretationen, wobei für die URIs des RDF-Vokabulars

```
rdf:type rdf:Property rdf:XMLLiteral rdf:nil  
rdf:List rdf:Statement rdf:subject rdf:predicate rdf:object  
rdf:first rdf:rest rdf:Seq rdf:Bag rdf:Alt  
rdf:_1 rdf:_2 ...
```

zusätzliche Forderungen gestellt werden,
die die intendierte Semantik der RDF-
Bezeichner realisieren:

RDF-INTERPRETATIONEN



Eine *RDF-Interpretation* für ein Vokabular V ist nun eine einfache Interpretation für das Vokabular $V \cup V_{\text{RDF}}$, welche zusätzlich folgende Bedingungen erfüllt:

- $x \in IP$ genau dann, wenn $\langle x, \text{rdf:Property}^{\mathcal{I}} \rangle \in I_{\text{EXT}}(\text{rdf:type}^{\mathcal{I}})$.
 x ist eine Property genau dann, wenn es mit der durch `rdf:Property` bezeichneten Ressource über die `rdf:type`-Property verbunden ist (dies führt auch automatisch dazu, dass für jede RDF-Interpretation $IP \subseteq IR$ gilt).
- wenn `"s"^^rdf:XMLLiteral` in V enthalten und s ein wohlgeformtes XML-Literal ist, dann
 - $I_L(\text{"s"^^rdf:XMLLiteral})$ ist der XML-Wert¹ von s ;
 - $I_L(\text{"s"^^rdf:XMLLiteral}) \in LV$;
 - $\langle I_L(\text{"s"^^rdf:XMLLiteral}), \text{rdf:XMLLiteral}^{\mathcal{I}} \rangle \in I_{\text{EXT}}(\text{rdf:type}^{\mathcal{I}})$
- wenn `"s"^^rdf:XMLLiteral` in V enthalten und s ein *nicht* wohlgeformtes XML-Literal ist, dann
 - $I_L(\text{"s"^^rdf:XMLLiteral}) \notin LV$ und
 - $\langle I_L(\text{"s"^^rdf:XMLLiteral}), \text{rdf:XMLLiteral}^{\mathcal{I}} \rangle \notin I_{\text{EXT}}(\text{rdf:type}^{\mathcal{I}})$.

RDF-INTERPRETATIONEN

- zusätzliche Forderung: jede RDF-Interpretation muss Modell der folgenden, „axiomatischen“ Tripel sein:

<code>rdf:type</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:subject</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:predicate</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:object</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:first</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:rest</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:value</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:_1</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:_2</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>...</code>		
<code>rdf:nil</code>	<code>rdf:type</code>	<code>rdf:List</code> .

RDF-FOLGERUNGEN



- automatische Folgerungen werden wieder über Ableitungsregeln realisiert:

$$\frac{}{u \ a \ x} \text{rdfax}$$

jedes axiomatische Tripel „u a x .“ kann immer abgeleitet werden

$$\frac{u \ a \ l \ .}{u \ a \ _ : n \ .} \text{lg}$$

Literals dürfen durch nicht anderweitig gebundene bnodes ersetzt werden

$$\frac{u \ a \ y \ .}{a \ \text{rdf:type} \ \text{rdf:Property} \ .} \text{rdf1}$$

für jedes Tripelprädikat kann abgeleitet werden dass es eine Entität aus der Klasse der Properties ist

$$\frac{u \ a \ l \ .}{_ : n \ \text{rdf:type} \ \text{rdf:XMLLiteral}} \text{rdf2}$$

wenn $_ : n$ durch lg dem wohlgeformten XML-Literal l zugewiesen wurde

RDF-FOLGERUNG

AIFB 

- Satz: Ein Graph G_2 RDF-folgt aus einem Graphen G_1 , wenn es einen Graphen G_1' gibt, so dass
 - G_1' aus G_1 via lg , $rdf1$, $rdf2$ und $rdfax$ hergeleitet werden kann und
 - G_2 aus G_1' einfach folgt.
- Beachte: zweistufiger Folgerungsprozess.

AGENDA



- Motivation
- Vorbetrachtungen
- einfache Folgerung
- RDF-Folgerung
- **RDFS-Folgerung**
- Unzulänglichkeiten von RDF(S)

RDFS-INTERPRETATIONEN

einfache Interpretationen

RDF-Interpretationen

RDFS-Interpretationen

- ...RDFS-Interpretationen sind spezielle RDF-Interpretationen, wobei für die URIs des RDFS-Vokabulars

```
rdfs:domain rdfs:range rdfs:Resource rdfs:Literal rdfs:Datatype  
rdfs:Class rdfs:subClassOf rdfs:subPropertyOf rdfs:member  
rdfs:Container rdfs:ContainerMembershipProperty rdfs:comment  
rdfs:seeAlso rdfs:isDefinedBy rdfs:label
```

zusätzliche Forderungen gestellt werden,
die die intendierte Semantik der RDF-
Bezeichner realisieren:

RDFS-INTERPRETATIONEN



Eine *RDFS-Interpretation* für ein Vokabular V ist eine RDF-Interpretation des Vokabulars $V \cup V_{\text{RDFS}}$, welche zusätzlich die folgenden Kriterien erfüllt:

- $IR = I_{\text{EXT}}(\text{rdfs:Resource}^{\mathcal{I}})$
Jede Ressource ist vom Typ `rdfs:Resource`.
- $LV = I_{\text{EXT}}(\text{rdfs:Literal}^{\mathcal{I}})$
Jedes ungetypte und jedes wohlgeformte getypte Literal ist vom Typ `rdfs:Literal`.
- Wenn $\langle x, y \rangle \in I_{\text{EXT}}(\text{rdfs:domain}^{\mathcal{I}})$ und $\langle u, v \rangle \in I_{\text{EXT}}(x)$,
dann $u \in I_{\text{EXT}}(y)$.
Ist x mit y durch die Property `rdfs:domain` verbunden und verbindet die Property x die Ressourcen u und v , dann ist u vom Typ y .
- Wenn $\langle x, y \rangle \in I_{\text{EXT}}(\text{rdfs:range}^{\mathcal{I}})$ und $\langle u, v \rangle \in I_{\text{EXT}}(x)$,
dann $v \in I_{\text{EXT}}(y)$.
Ist x mit y durch die Property `rdfs:range` verbunden und verbindet die Property x die Ressourcen u und v , dann ist v vom Typ y .
- $I_{\text{EXT}}(\text{rdfs:subPropertyOf}^{\mathcal{I}})$ ist reflexiv und transitiv auf IP .
Die `rdfs:subPropertyOf`-Property verbindet jede Property mit sich selbst.
Darüber hinaus gilt: Verbindet `rdfs:subPropertyOf` die Property x mit Property y und außerdem y mit der Property z , so verbindet `rdfs:subPropertyOf` auch x direkt mit z .

RDFS-INTERPRETATIONEN



- Wenn $\langle x, y \rangle \in I_{\text{EXT}}(\text{rdfs:subPropertyOf}^{\mathcal{I}})$,
dann $x, y \in IP$ und $I_{\text{EXT}}(x) \subseteq I_{\text{EXT}}(y)$.
Wird x mit y durch `rdfs:subPropertyOf` verbunden, dann sind sowohl x als auch y Property's und jedes in der Extension von x enthaltene Ressourcenpaar ist auch in der Extension von y enthalten.
- Wenn $x \in IC$,
dann $\langle x, \text{rdfs:Resource}^{\mathcal{I}} \rangle \in I_{\text{EXT}}(\text{rdfs:subClassOf}^{\mathcal{I}})$.
Bezeichnet x eine Klasse, dann muss es eine Unterklasse der Klasse aller Ressourcen sein, d.h., das Paar aus x und `rdfs:Resource` ist in der Extension von `rdfs:subClassOf`.
- Wenn $\langle x, y \rangle \in I_{\text{EXT}}(\text{rdfs:subClassOf}^{\mathcal{I}})$,
dann $x, y \in IC$ und $I_{\text{CEXT}}(x) \subseteq I_{\text{CEXT}}(y)$.
Stehen x und y in der `rdfs:subClassOf`-Beziehung, sind sowohl x als auch y Klassen und die (Klassen-)Extension von x ist Teilmenge der (Klassen-)Extension von y .
- $I_{\text{EXT}}(\text{rdfs:subClassOf}^{\mathcal{I}})$ ist reflexiv und transitiv auf IC .
Die `rdfs:subClassOf`-Property verbindet jede Klasse mit sich selbst. Darüber hinaus folgt, wann immer diese Property Klasse x mit Klasse y und Klasse y mit Klasse z verbindet, dass sie x auch direkt mit z verbindet.

RDFS-INTERPRETATIONEN



- Wenn $x \in I_{\text{EXT}}(\text{rdfs:ContainerMembershipProperty}^{\mathcal{I}})$,
dann $\langle x, \text{rdfs:member}^{\mathcal{I}} \rangle \in I_{\text{EXT}}(\text{rdfs:subPropertyOf}^{\mathcal{I}})$.
Ist x eine Property vom Typ `rdfs:ContainerMembershipProperty`,
so steht sie in der `rdfs:subPropertyOf`-Beziehung zur `rdfs:member-`
Property.
- Wenn $x \in I_{\text{EXT}}(\text{rdfs:Datatype}^{\mathcal{I}})$,
dann $\langle x, \text{rdfs:Literal}^{\mathcal{I}} \rangle \in I_{\text{EXT}}(\text{rdfs:subClassOf}^{\mathcal{I}})$
Ist ein x als Element der Klasse `rdfs:Datatype` „getypt“, dann muss
dieses auch eine Unterklasse der Klasse aller Literalwerte (bezeichnet
mit `rdfs:Literal`) sein.

- ...dazu kommen dann noch jede Menge
weitere axiomatische Tripel:

RDFS-INTERPRETATIONEN



rdf:type	rdfs:domain	rdfs:Resource .	rdfs:ContainerMembershipProperty		
rdfs:domain	rdfs:domain	rdf:Property .		rdfs:subClassOf	rdf:Property .
rdfs:range	rdfs:domain	rdf:Property .	rdf:Alt	rdfs:subClassOf	rdfs:Container .
rdfs:subPropertyOf	rdfs:domain	rdf:Property .	rdf:Bag	rdfs:subClassOf	rdfs:Container .
rdfs:subClassOf	rdfs:domain	rdfs:Class .	rdf:Seq	rdfs:subClassOf	rdfs:Container .
rdf:subject	rdfs:domain	rdf:Statement .			
rdf:predicate	rdfs:domain	rdf:Statement .	rdfs:isDefinedBy	rdfs:subPropertyOf	rdfs:seeAlso .
rdf:object	rdfs:domain	rdf:Statement .			
rdfs:member	rdfs:domain	rdfs:Resource .	rdf:XMLLiteral	rdf:type	rdfs:Datatype .
rdf:first	rdfs:domain	rdf:List .	rdf:XMLLiteral	rdfs:subClassOf	rdfs:Literal .
rdf:rest	rdfs:domain	rdf:List .	rdfs:Datatype	rdfs:subClassOf	rdfs:Class .
rdfs:seeAlso	rdfs:domain	rdfs:Resource .			
rdfs:isDefinedBy	rdfs:domain	rdfs:Resource .	rdf:_1	rdf:type	
rdfs:comment	rdfs:domain	rdfs:Resource .			rdfs:ContainerMembershipProperty .
rdfs:label	rdfs:domain	rdfs:Resource .	rdf:_1	rdfs:domain	rdfs:Resource .
rdf:value	rdfs:domain	rdfs:Resource .	rdf:_1	rdfs:range	rdfs:Resource .
			rdf:_2	rdf:type	
rdf:type	rdfs:range	rdfs:Class .			rdfs:ContainerMembershipProperty .
rdfs:domain	rdfs:range	rdfs:Class .	rdf:_2	rdfs:domain	rdfs:Resource .
rdfs:range	rdfs:range	rdfs:Class .	rdf:_2	rdfs:range	rdfs:Resource .
rdfs:subPropertyOf	rdfs:range	rdf:Property		
rdfs:subClassOf	rdfs:range	rdfs:Class .			
rdf:subject	rdfs:range	rdfs:Resource .			
rdf:predicate	rdfs:range	rdfs:Resource .			
rdf:object	rdfs:range	rdfs:Resource .			
rdfs:member	rdfs:range	rdfs:Resource .			
rdf:first	rdfs:range	rdfs:Resource .			
rdf:rest	rdfs:range	rdf:List .			
rdfs:seeAlso	rdfs:range	rdfs:Resource .			
rdfs:isDefinedBy	rdfs:range	rdfs:Resource .			
rdfs:comment	rdfs:range	rdfs:Literal .			
rdfs:label	rdfs:range	rdfs:Literal .			
rdf:value	rdfs:range	rdfs:Resource .			

RDFS-FOLGERUNG

AIFB 

- automatische Folgerungen werden wieder über Ableitungsregeln realisiert:

$$\begin{array}{l}
 \frac{}{u \ a \ x} \text{ rdfsax} \qquad \frac{u \ \text{rdfs:subPropertyOf} \ v \ . \quad v \ \text{rdfs:subPropertyOf} \ x \ .}{u \ \text{rdfs:subPropertyOf} \ x \ .} \text{ rdfs5} \qquad \frac{u \ \text{rdf:type} \ \text{rdfs:ContainerMembershipProperty} \ .}{u \ \text{rdfs:subPropertyOf} \ \text{rdfs:member} \ .} \text{ rdfs12} \\
 \\
 \frac{u \ a \ _ : n \ .}{u \ a \ l \ .} \text{ gl} \qquad \frac{u \ \text{rdf:type} \ \text{rdf:Property} \ .}{u \ \text{rdfs:subPropertyOf} \ u \ .} \text{ rdfs6} \qquad \frac{u \ \text{rdf:type} \ \text{rdfs:Datatype} \ .}{u \ \text{rdfs:subClassOf} \ \text{rdfs:Literal} \ .} \text{ rdfs13} \\
 \\
 \frac{u \ a \ l \ .}{_ : n \ \text{rdf:type} \ \text{rdfs:Literal} \ .} \text{ rdfs1} \qquad \frac{a \ \text{rdfs:subPropertyOf} \ b \ . \quad u \ a \ y \ .}{u \ b \ y \ .} \text{ rdfs7} \\
 \\
 \frac{a \ \text{rdfs:domain} \ x \ . \quad u \ a \ y \ .}{u \ \text{rdf:type} \ x \ .} \text{ rdfs2} \qquad \frac{u \ \text{rdf:type} \ \text{rdfs:Class} \ .}{u \ \text{rdfs:subClassOf} \ \text{rdfs:Resource} \ .} \text{ rdfs8} \\
 \\
 \frac{a \ \text{rdfs:range} \ x \ . \quad u \ a \ v \ .}{v \ \text{rdf:type} \ x \ .} \text{ rdfs3} \qquad \frac{u \ \text{rdfs:subClassOf} \ x \ . \quad v \ \text{rdf:type} \ u \ .}{v \ \text{rdf:type} \ x \ .} \text{ rdfs9} \\
 \\
 \frac{u \ a \ x \ .}{u \ \text{rdf:type} \ \text{rdfs:Resource} \ .} \text{ rdfs4a} \qquad \frac{u \ \text{rdf:type} \ \text{rdfs:Class} \ .}{u \ \text{rdfs:subClassOf} \ u \ .} \text{ rdfs10} \\
 \\
 \frac{u \ a \ v \ .}{v \ \text{rdf:type} \ \text{rdfs:Resource} \ .} \text{ rdfs4b} \qquad \frac{u \ \text{rdfs:subClassOf} \ v \ . \quad v \ \text{rdfs:subClassOf} \ x \ .}{u \ \text{rdfs:subClassOf} \ x \ .} \text{ rdfs11}
 \end{array}$$

RDFS-FOLGERUNG



- wichtige Definition: XML-Clash

```
ex:hatSmiley          rdfs:range      rdf:Literal .  
ex:böseBemerkung      ex:hatSmiley    ">:->"^^XMLLiteral .
```

- tritt auf, wenn einem Knoten vom Typ `rdf:Literal` ein nicht-wohlgeformter Literalwert zugewiesen werden muss.

RDFS-FOLGERUNG

AIFB 

- Satz: Ein Graph RDFS-folgt aus G_1 genau dann, wenn es einen Graphen G_1' gibt, der durch Anwendung der Regeln lg, gl, rdfax, rdf1, rdf2, rdfs1 – rdfs13 und rdfsax aus G_1 folgt, so dass
 - G_2 aus G_1' einfach folgt oder
 - G_1' einen XML-Clash enthält.

AGENDA



- Motivation
- Vorbetrachtungen
- einfache Folgerung
- RDF-Folgerung
- RDFS-Folgerung
- Unzulänglichkeiten von RDF(S)

WAS KANN RDF(S) NICHT?

- bestimmte (vernünftig) scheinende Folgerungen können nicht RDFS-gefolgert werden, z.B.

```
ex:sprichtMit    rdfs:domain    ex:Homo .
ex:Homo         rdfs:subClassOf ex:Primates .
```

impliziert

```
ex:sprichtMit    rdfs:domain    ex:Primates .
```

- mögliche Lösung: noch stärkere, „extensionale“, Semantik
- keine Möglichkeit, Negation auszudrücken